

CPP05 – EX01 : DOCUMENT PÉDAGOGIQUE POUR DÉBUTANTS

INTRODUCTION

Cet exercice introduit la programmation orientée objet en C++ à travers deux classes : Bureaucrat et Form.

Il permet d'enseigner trois concepts fondamentaux :

- La notion de classe et d'objet
- Les interactions entre objets
- Les exceptions pour gérer les erreurs

SECTION 1 : LE CONCEPT DE GRADE (1 À 150)

Chaque bureaucrate possède un grade compris entre 1 et 150.

1 = meilleur niveau, 150 = pire niveau.

Tout grade < 1 ou > 150 est interdit.

Cas concret valide :

```
Bureaucrat alice("Alice", 10);
```

→ Grade accepté.

Cas contraire invalide :

```
Bureaucrat bob("Bob", 200);
```

→ Exception levée : GradeTooLowException.

SECTION 2 : LA CLASSE BUREAUCRAT

Un Bureaucrat a :

- un nom (constant)
- un grade (modulable mais validé)

Il peut améliorer ou dégrader son grade, dans les limites autorisées.

Cas concret valide :

```
Bureaucrat c("Charlie", 10);
```

```
c.incrementGrade(); // devient 9
```

Cas contraire :

```
Bureaucrat d("Dora", 1);  
d.incrementGrade(); // impossible, exception
```

SECTION 3 : LA CLASSE FORM

Un Form possède :

- `_name` (const)
- `_isSigned` (booléen)
- `_gradeToSign` (const)
- `_gradeToExecute` (const)

Les valeurs doivent être entre 1 et 150.

Cas concret valide :

```
Form f("TopSecret", 20, 5);
```

Cas contraire :

```
Form bad("BadForm", 0, 10);  
→ exception : GradeTooHighException.
```

SECTION 4 : SIGNATURE D'UN FORMULAIRE

Un bureaucrate peut signer un formulaire seulement si son grade est suffisamment bon.

Règle :

```
bureaucrat.getGrade() <= form.getGradeToSign()
```

Cas concret valide :

```
Alice (grade 10) signe TopSecret (grade 20)  
→ succès, formulaire signé.
```

Cas contraire :

Bob (grade 150) tente de signer ImportantPaper (grade 100)

→ exception, signature refusée.

SECTION 5 : UTILISATION DES EXCEPTIONS

Les exceptions permettent de signaler une action impossible.

Elles évitent les crashes et rendent le code plus robuste.

Exemple valide :

```
try { Bureaucrat b("Bob", 200); }  
catch (...) { afficher erreur }
```

Exemple contraire :

Sans try/catch → crash du programme.

CONCLUSION

Cet exercice permet d'enseigner :

- La structure d'une classe
- Les restrictions et validations
- La communication entre objets
- L'usage pratique des exceptions
- La conception orientée objet

C'est un excellent support pour enseigner les bases du C++ à un débutant en illustrant chaque notion par des cas concrets et leurs opposés.